L 11

```
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS     CCCCCCCCCCCC     AAAAAAAAA     LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS     CCCCCCCCCCCC     AAAAAAAAA     LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS     CCCCCCCCCCCC     AAAAAAAAA     LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     CCC              AAA         AAA  LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     CCC              AAA         AAA  LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     CCC              AAA         AAA  LLL
PPP              AAAAAAAAAAAAAAA              SSS  CCC              AAAAAAAAAAAAAAA  LLL
PPP              AAAAAAAAAAAAAAA              SSS  CCC              AAAAAAAAAAAAAAA  LLL
PPP              AAAAAAAAAAAAAAA              SSS  CCC              AAAAAAAAAAAAAAA  LLL
PPP              AAA         AAA              SSS  CCC              AAA         AAA  LLL
PPP              AAA         AAA              SSS  CCC              AAA         AAA  LLL
PPP              AAA         AAA              SSS  CCC              AAA         AAA  LLL
PPP              AAA         AAA  SSSSSSSSSSSS        CCCCCCCCCCCC  AAA         AAA  LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS        CCCCCCCCCCCC  AAA         AAA  LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS        CCCCCCCCCCCC  AAA         AAA  LLLLLLLLLLLLLLL
```

```
                                              B  4
**FILE**ID**PASRT1



          PPPPPPPP      AAAAAA      SSSSSSSS   RRRRRRR    TTTTTTTTTT      11
          PPPPPPPP      AAAAAA      SSSSSSSS   RRRRRRR    TTTTTTTTTT      11
          PP      PP  AA      AA  SS          RR      RR      TT        1111
          PP      PP  AA      AA  SS          RR      RR      TT        1111
          PP      PP  AA      AA  SS          RR      RR      TT          11
          PP      PP  AA      AA  SS          RR      RR      TT          11
          PPPPPPPP    AA      AA    SSSSSS    RRRRRRRR        TT          11
          PPPPPPPP    AA      AA    SSSSSS    RRRRRRRR        TT          11
          PP          AAAAAAAAAA          SS  RR  RR          TT          11
          PP          AAAAAAAAAA          SS  RR  RR          TT          11
          PP          AA      AA          SS  RR    RR        TT          11          ....
          PP          AA      AA          SS  RR    RR        TT          11          ....
          PP          AA      AA  SSSSSSSS    RR      RR      TT        111111        ....
          PP          AA      AA  SSSSSSSS    RR      RR      TT        111111        ....


          LL            IIIIII      SSSSSSSS
          LL            IIIIII      SSSSSSSS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II        SSSSSS
          LL              II        SSSSSS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LLLLLLLLLL    IIIIII    SSSSSSSS
          LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
    0000  ;
    0000  ;*********************************************************************
    0000  ;*                                                                   *
    0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
    0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
    0000  ;*  ALL RIGHTS RESERVED.                                             *
    0000  ;*                                                                   *
    0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
    0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
    0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
    0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
    0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
    0000  ;*  TRANSFERRED.                                                     *
    0000  ;*                                                                   *
    0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
    0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
    0000  ;*  CORPORATION.                                                     *
    0000  ;*                                                                   *
    0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
    0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.          *
    0000  ;*                                                                   *
    0000  ;*                                                                   *
    0000  ;*********************************************************************
    0000  ;*                                                                   *
    0000  ;*                         PAS$RT_UTIL                               *
    0000  ;*            RUNTIME SUPPORT MODULE FOR PASCAL -- SECTION 1         *
    0000  ;*                                                                   *
    0000  ;*                  VERSION V1.0-1 -- OCTOBER 1979                   *
    0000  ;*                                                                   *
    0000  ;*   This module defines the following routines:                     *
    0000  ;*                                                                   *
    0000  ;*      pas$entry:   JSB routine to expand stack on procedure entry  *
    0000  ;*      pas$unwind:  routine to unwind stack for nonlocal goto       *
    0000  ;*      pas$clock:   routine to implement the Pascal function clock  *
    0000  ;*      pas$card:    routine to implement the Pascal function card   *
    0000  ;*      pas$getargs: routine to get compiler options settings        *
    0000  ;*      pas$extract,                                                 *
    0000  ;*      pas$insert:  routines for compile time variable field handling  *
    0000  ;*                                                                   *
    0000  ;*   Written by: Jeff Scofield 10-Dec-78                             *
    0000  ;*               Hellmut Golde 15-Feb-79                             *
    0000  ;*               Jan Sanislo   22-Feb-79                             *
    0000  ;*                                                                   *
    0000  ;*   Edit History:
    0000  ;*      01-002: Eliminate body of pas$entry for VMS V2.0.  Leave the entry
    0000  ;*              point available for compatibility with older versions of
    0000  ;*              of the compiler.
    0000  ;*                                   Paul Hohensee   21FEB80
    0000  ;*
    0000  ;       01-003: Multiply result of PAS$CLOCK by 10.
    0000  ;                                    Paul Hohensee   20-Jul-81
    0000  ;
    0000  ;*********************************************************************
    0000          .title  pas$rt_util
    0000          .IDENT  'V04-000'
00000000          .psect _pas$code,pic,shr,exe,nowrt
    0000  ;
```

```
                            0000  ;       ROUTINE TO EXPAND STACK WHEN NECESSARY UPON PROCEDURE ENTRY
                            0000  ;
                            0000  pas$entry::
                        05  0000          rsb                   ; leave entry point for compatibility
                            0001  ;
                            0001  ;
                            0001  ;       ROUTINE TO IMPLEMENT THE PROCEDURE PAS$UNWIND
                            0001  ;
                            0001  ;       Modified 5/22/79 - Restore correct SP in case of pathological goto
                            0001  ;                               Jan Sanislo
                            0001  ;
                      0000  0001          .entry  pas$unwind,^m<>
           0C AD    50  D1  0003  loop:   cmpl    r0,12(fp)
                    06  13  0007          beql    lastret
        10 AD    F7 AF  DE  0009          moval   loop,16(fp)
                        04  000E          ret
                            000F  lastret:
   10 AD    00000018'EF  9E  000F          movab   fixsp,16(fp)
                        04  0017          ret
                            0018  ;
           5E    F4 AD  D0  0018  fixsp:  movl    -12(fp),sp
                    61  17  001C          jmp     (r1)
                            001E  ;
                            001E  ;       ROUTINE TO IMPLEMENT THE PASCAL FUNCTION CLOCK
                            001E  ;
                            001E          $jpidef
                      0000  001E          .entry  pas$clock,^m<>
                    00  DD  0020          pushl   #0                            ; make room for returned cpu time
                            0022  ;
                            0022  ;       Create request list on stack at -20(fp)
                            0022  ;
                    7E  7C  0022          clrq    -(sp)                         ; two zero longwords
                 FC AD  DF  0024          pushal  -4(fp)                        ; address of spot to get cpu time
           04070004 8F  DD  0027          pushl   #<jpi$_cputim@16>!4           ; size and request words
                            002D  ;
                            002D  ;       Push arguments and call sys$getjpi
                            002D  ;
                    7E  7C  002D          clrq    -(sp)                         ; arg6,arg7--null arguments
                    00  DD  002F          pushl   #0                            ; arg5--null argument
                 EC AD  DF  0031          pushal  -20(fp)                       ; arg4--address of request list
                    7E  7C  0034          clrq    -(sp)                         ; arg2,arg3--null arguments
                    00  DD  0036          pushl   #0                            ; arg1--null argument
   00000000'GF      07  FB  0038          calls   #7,G^sys$getjpi               ; get cpu time from system
        50    0A    FC AD  C5  003F          MULL3   -4(FP),#10,R0                 ; multiply by 10 to get milliseconds
                        04  0044          ret
                            0045  ;
                            0045  ;       ROUTINE TO IMPLEMENT THE PASCAL FUNCTION CARD
                            0045  ;
                      003C  0045          .entry  pas$card,^m<r2,r3,r4,r5>
                    50  D4  0047          clrl    r0                            ; clear return count
                    51  D4  0049          clrl    r1                            ; clear starting position
                    55  D4  004B          clrl    r5                            ; clear size comparison reg.
           52    04 AC  D0  004D          movl    4(ap),r2                      ; get length of set
               53   52  D0  0051          movl    r2,r3                         ; into 3 registers
               54   52  D0  0054  10$:    movl    r2,r4                         ;
               20   54  D1  0057          cmpl    r4,#32                        ; check size field
                    06  15  005A          bleq    20$                           ; is ok.
```

```
                  54  20    D0 005C                movl    #32,r4                  ; otherwise set size to 32
                  52  20    C2 005F                subl2   #32,r2                  ;
                  55  54    C0 0062     20$:       addl2   r4,r5                   ; increment size comparison
      51   08 BC  54  51    EA 0065     30$:       ffs     r1,r4,@8(ap),r1         ; find next "1" bit
                      0A    13 006B                beql    40$                     ; done if Z-bit = 1
                      50    D6 006D                incl    r0                      ; increment count
                      51    D6 006F                incl    r1                      ; increment starting position
             54   55  51    C3 0071                subl3   r1,r5,r4                ; compute new length
                      EE    11 0075                brb     30$                     ; loop until done
                  51  53    D1 0077     40$:       cmpl    r3,r1                   ; check if done
                      D8    14 007A                bgtr    10$                     ;
                            04 007C                ret                             ; return to caller
                               007D        ;
                               007D        ;    ROUTINE TO GET OPTION SETTINGS FROM COMMAND LINE
                               007D        ;
                               007D        ;    This routine gets the option settings from the command line, which
                               007D        ;    were passed as arguments to the main program level.  These argu-
                               007D        ;    ments are not available within Pascal, which is why this routine is
                               007D        ;    required.  This routine must be called directly from the main
                               007D        ;    program, as it assumes that the main program's saved AP is on the
                               007D        ;    stack at 8(FP).
                               007D        ;
                          0100 007D                .entry  pas$getargs,^m<r8>
                  58    08 AD   D0 007F            movl    8(fp),r8                ; r8 <-- saved ap of main program
            04 BC       04 B8   B0 0083            movw    @4(r8),@4(ap)           ; set first return parameter
            08 BC       08 B8   B0 0088            movw    @8(r8),@8(ap)           ; set second return parameter
            0C BC       0C A8   D0 008D            movl    12(r8),@12(ap)          ; set third return parameter
       10 BC    10 B8     0B    28 0092            movc3   #11,@16(r8),@16(ap)     ; set fourth return parameter
       14 BC    14 B8     0B    28 0098            movc3   #11,@20(r8),@20(ap)     ; set fifth return parameter
   18 BC    18 B8     0080 8F   28 009E            movc3   #128,@24(r8),@24(ap)    ; set sixth return parameter
            1C BC       1C A8   D0 00A6            movl    28(r8),@28(ap)          ; set seventh return parameter.
                                04 00AB            ret                             ; return to caller
                               00AC        ;
                               00AC        ;    ROUTINES FOR VARIABLE FIELD INSERTION AND EXTRACTION BY COMPILER
                               00AC        ;
                          0000 00AC                .entry  pas$insert,^m<>
04 BC    10 AC    0C AC    08 BC   F0 00AE         insv    @8(ap),12(ap),16(ap),@4(ap)
                                04 00B7            ret
                               00B8        ;
                          0000 00B8                .entry  pas$extract,^m<>
08 BC    04 BC    10 AC    0C AC   EF 00BA         extzv   12(ap),16(ap),@4(ap),@8(ap)
                                04 00C3            ret
                               00C4                .end
```

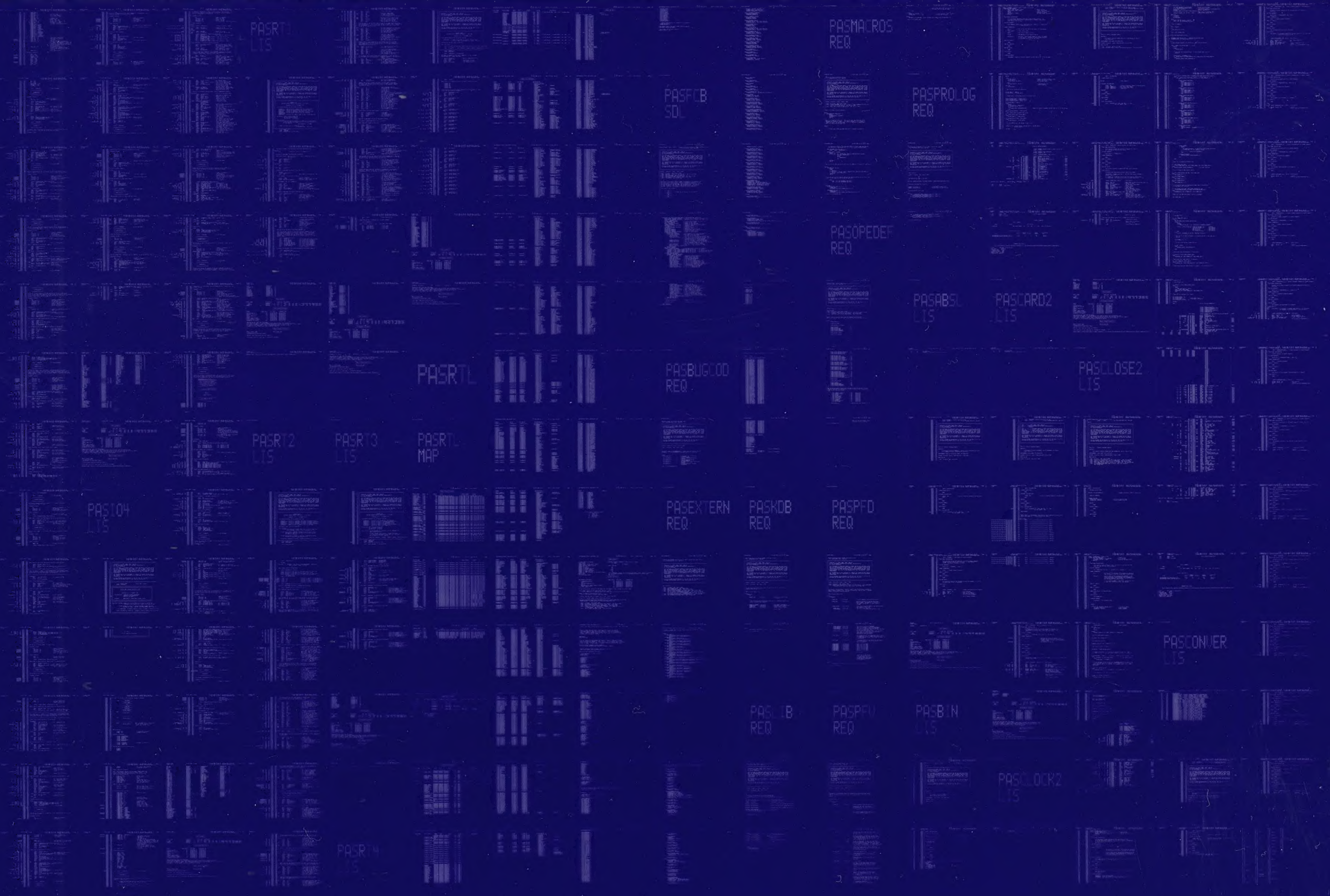
```
FIXSP                            00000018 R      01
JPI$_CPUTIM                    = 00000407
LASTRET                          0000000F R      01
LOOP                             00000003 R      01
PAS$CARD                         00000045 RG     01
PAS$CLOCK                        0000001E RG     01
PAS$ENTRY                        00000000 RG     01
PAS$EXTRACT                      000000B8 RG     01
PAS$GETARGS                      0000007D RG     01
PAS$INSERT                       000000AC RG     01
PAS$UNWIND                       00000001 RG     01
SYS$GETJPI                       ********    X   01

                                              +----------------+
                                              ! Psect synopsis !
                                              +----------------+

PSECT name                       Allocation            PSECT No.  Attributes
----------                       ----------            ---------  ----------
.  ABS  .                        00000000  (     0.)   00 (  0.)  NOPIC    USR     CON     ABS     LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
_PAS$CODE                        000000C4  (   196.)   01 (  1.)    PIC    USR     CON     REL     LCL   SHR   EXE   RD   NOWRT NOVEC BYTE
$ABS$                            00000000  (     0.)   02 (  2.)  NOPIC    USR     CON     ABS     LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                          +------------------------+
                                          ! Performance indicators !
                                          +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      34    00:00:00.09    00:00:00.90
Command processing                 135    00:00:00.50    00:00:02.14
Pass 1                             128    00:00:01.66    00:00:03.55
Symbol table sort                    0    00:00:00.10    00:00:00.09
Pass 2                              45    00:00:00.53    00:00:01.10
Symbol table output                  3    00:00:00.02    00:00:00.03
Psect synopsis output                3    00:00:00.02    00:00:00.02
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               351    00:00:02.92    00:00:07.84

The working set limit was 1050 pages.
7791 bytes (16 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 120 non-local and 4 local symbols.
157 source lines were read in Pass 1, producing 28 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

                                         +--------------------------+
                                         ! Macro library statistics !
                                         +--------------------------+

Macro library name                       Macros defined
------------------                       --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     4

168 GETS were required to define 4 macros.

There were no errors, warnings or information messages.
```

```
                                                    G  4
PAS$RT_UTIL                                                     16-SEP-1984 02:08:46  VAX/VMS Macro V04-00        Page   5
VAX-11 Macro Run Statistics                                      5-SEP-1984 02:32:39  [PASCAL.SRC]PASRT1.MAR;1           (1)

MACRO/DISABLE=TRACE/LIS=LIS$:PASRT1/OBJ=OBJ$:PASRT1 MSRC$:PASRT1/UPDATE=(ENH$:PASRT1)
```

```
PAS
VAX

Ass

The
133
The
528
8 p



Mac
---
_$2

91

The

MAC
```

0293 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



PASRT1
LIS

PASMACROS
REQ

PASFCB
SDL

PASPROLOG
REQ

PASOPEDEF
REQ

PASABSL
LIS

PASCARD2
LIS

PASRTL

PASBUGCOD
REQ

PASCLOSE2
LIS

PASRT2
LIS

PASRT3
LIS

PASRTL
MAP

PASIO4
LIS

PASEXTERN
REQ

PASKDB
REQ

PASPFD
REQ

PASCONVER
LIS

PASLIB
REQ

PASPFV
REQ

PASBIN
LIS

PASCLOCK2
LIS

PASRT4
LIS